

Patent claims

1. A method for managing and transmitting events from
5 a server (2) via a communication link (9) to at least
one client (1), where
 - possible events are logged in a client event
service (6) for the purpose of initializing and/or
updating the client (1),
 - 10 - possible events are logged in a server event
service (7) for the purpose of initializing and/or
updating the server (2),
 - detected events which have been logged are
transferred from an installation interface (10) to
15 the server event service (7),
 - requests initiated by the client event service (6)
regarding the detected events are made to the
server event service (7),
 - on the basis of a request which has been made to
20 the server event service (7) the detected events
are transmitted to the client event service (6),
and
 - events received by the client event service (6)
are transmitted to a client application (4).
- 25 2. The method as claimed in claim 1, characterized in
that the events to be transmitted are detected by a
data capture unit in a technical installation (3) and
are reported to the installation interface (10) of the
30 server (2).
3. The method as claimed in either of claims 1 and 2,
characterized in that the client application (4) logs a
client callback function (41) in the client event
35 service (6) for every event about which it is to be
notified, and the client event service (6) uses the
communication link (9) to log a corresponding server
callback function (72) in the server event service (7).

4. The method as claimed in claim 3, characterized in that to log the callback functions (41, 72) for an event with which the same event name is associated with the client (1) and with the server (2) in preparation for the method, the following steps are performed:

- the client application (4) calls a client logging function (61) from the client event service (6) and provides said function with the name of the event in question and with a pointer to the client callback function (41) which is to be logged,
- the client logging function (61) generates a unique event identifier and transmits the event identifier and the event name via the communication link (9) to a server logging function (71) of the server event service (7),
- the server logging function (71) logs a server callback function (72) with the installation interface (10) by transferring the event name,
- the server logging function (71) stores a data record, which contains at least the event identifier and a pointer to the server callback function (72) which is to be logged, in a server event table (74),
- the server logging function (71) reports the performance of the logging operation to the client logging function (61) of the client event service (6) via the communication link (9), and
- the client logging function (61) logs the client callback function (41) by storing a data record in a client event table (62), the data record containing at least the event identifier and a pointer to the client callback function (41) which is to be logged.

5. The method as claimed in claim 4, characterized in that after a client callback function (41) has been logged for the first time the client logging function

(61) starts a request generator (63) which then makes requests for event transmission to the server event service (7).

5 6. The method as claimed in claim 5, characterized in that the request generator (63) of the client event service (6) makes the requests for event transmission to the server event service (7) cyclically.

10 7. The method as claimed in one of claims 3 to 6, characterized in that events are transmitted by performing the following steps:

- the installation interface (10) detects an event which has occurred and calls the server callback function (72) logged for this event,
- 15 - the server callback function (72) produces an entry describing the event in at least one event queue (75),
- upon the next request from the client event service (6) for event transmission the server event service (7) reads the entry produced from the event queue (75) and transmits it via the communication link (9) to the client event service (6),
- 20 - the client event service (6) takes the entry received and ascertains and calls the client callback function (41) logged for this event, and
- the client callback function (41) executes a defined action for the corresponding event in the client application (4).
- 30

8. The method as claimed in claim 7, characterized in that optionally a tidying function (77) of the server event service (7) is called which deletes the server event table (74) and the event queue (75) if the client event service (6) is no longer communicating with the server event service (7).

35

9. A system for managing and transmitting events from a server (2) via a communication link (9) to at least one client (1), where

- 5 - for the purpose of logging possible events the client (1) has at least one client event service (6) which uses a communication link (9) to make requests for event transmission to a server event service (7) and to transmit received events to a client application (4),
- 10 - for the purpose of logging possible events the server (2) has at least one server event service (7) which uses a communication link (9) to transmit events to a client event service (6),
- 15 - the server (2) has at least one installation interface (10) which transfers events which have occurred to the at least one server event service (7).

20 10. The system as claimed in claim 9, characterized in that the installation interface (10) is connected to a data capture unit of a technical installation (3) in order to read in events detected by the data capture unit.

25 11. The system as claimed in either of claims 9 and 10, characterized in that the server event service (7) has at least one server callback function (72) which can be logged for at least one event and which is called when an event for which it is logged occurs.

30 12. The system as claimed in one of claims 9 to 11, characterized in that the server event service (7) has at least one server logging function (71) for logging server callback functions (72), at least one server event table (74) for holding data records which describe a respective logging operation, and at least
35 one event queue (75) for holding entries which describe a respective event.

13. The system as claimed in one of claims 9 to 12, characterized in that the server event service (7) has, for every client event service (6) with which it communicates via a communication link (9), a separate client data record (73) which respectively contains at least one server event table (74) and at least one event queue (75).

14. The system as claimed in claim 13, characterized in that the server event service (7) has a tidying function (77) which deletes the client data record (73) if the associated client event service (6) is no longer communicating with the server event service (7).

15. The system as claimed in one of claims 12 to 14, characterized in that the server event table (74) is in the form of a hash table and holds data records which contain at least one event identifier and a pointer to a server callback function (72) which is to be logged.

16. The system as claimed in one of claims 9 to 15, characterized in that the client (1) has at least one client callback function (41) which can be logged for at least one event and which is called when the event for which it is logged occurs.

17. The system as claimed in one of claims 9 to 16, characterized in that the client event service (6) has at least one client logging function (61) for logging client callback functions (41), at least one client event table (62) for holding data records which describe the log, and at least one request generator (63) for making cyclic requests for event transmission.

18. The system as claimed in claim 17, characterized in that the client event table (62) is in the form of a hash table and holds data records which contain at

least one event identifier and a pointer to a client
callback function (41) which is to be logged.